

# HIGH-LEVEL PHYSICS APPLICATION FOR THE EMITTANCE MEASUREMENT BY ALLISON SCANNER\*

T. Zhang<sup>†</sup>, S. M. Lund, T. Maruta, J. C. Wong  
Facility for Rare Isotope Beams, Michigan State University, East Lansing, MI, USA

## Abstract

On the ion accelerator, transverse emittance diagnostics usually happens at the low-energy transportation region, one device named ‘Allison scanner’ is commonly used to achieve this goal. In this contribution, we present the software development for both the high-level GUI application and the online data analysis, to help the users to get the beam emittance information as precise and efficient as possible, meanwhile, the entire workflow including the UI interaction would be smooth and friendly enough. One soft-IOC application has been created for the device simulation and application development. A dedicated 2D image data visualization widget is also introduced for the general-purposed GUI application development with PyQt5.

## INTRODUCTION

The driver LINAC of Facility for Rare Isotope Beams (FRIB) is designed to be able to accelerate a range of stable ions from Oxygen to Uranium to the kinetic energy > 200 MeV/u, with a final delivered beam power at the target of up to 400 kW, which is more than two orders advancement in the heavy ion accelerators regime [1, 2].

The transverse beam emittance is usually diagnosed at the lower beam energy transportation line (LEBT), the projected two-dimensional phase space distribution could be sampled by moving the step motor on which is mounted with an open slit to capture the beam at the different transverse location, at the meantime, sweeping the deflection voltage applied onto the parallelly displaced electric plates to figure out the incident angle. The voltage sweeping ( $V_0$ ) could map the transverse divergence  $x'$  information, by the simple formula  $x' = \frac{qV_0L}{2dE_k}$ , where  $V_0$  is the voltage applied onto the upper (+ $V_0$ ) and bottom ( $-V_0$ ) plates,  $L$  is the effective horizontal displacement of the deflection and  $E_k$  is the kinetic energy of the incidental ion. Such kind of emittance measurement device is so-called ‘Allison scanner’, which was proposed in early last 80’s [3].

Reference [4] investigates the details about the data analysis about the emittance measurement by Allison scanner at FRIB LEBT beamline. And also presents the developed Python routine for quick data post-processing. In this paper, we present the continued software development for Allison scanner, the deployed high-level control software application features the core data analysis algorithm developed (see the appendix section of ref [4]), and more user-friendly

interactive data processing is introduced, as well as other subtle improvement that can boost the efficiency and user experience.

## ALLISON SCANNER APPLICATION

The physics high-level applications for FRIB are systematically designed and developed. Started from the Python interactive scripting environment, to the higher-level GUI applications, various techniques are utilized to make the development robust and efficient, as well as the deployment to FRIB’s controls network [5].

Allison scanner is one of the ‘slow’ devices used by FRIB LINAC, ‘slow’ means the response of the data for process from the device usually happens a few seconds or minutes after the trigger command is sent out. The EPICS [6] hardware input output controller (IOC) is developed by diagnostic engineers, which is responsible for the device controls, that is to make the step motor and the voltage sweeping as requested by controlling specific controls process variables (PVs). At FRIB, there also comes with a dedicate CS-studio OPI for the user to operate the Allison scanner device, rather than typing the tedious commands to talk to the IOC.

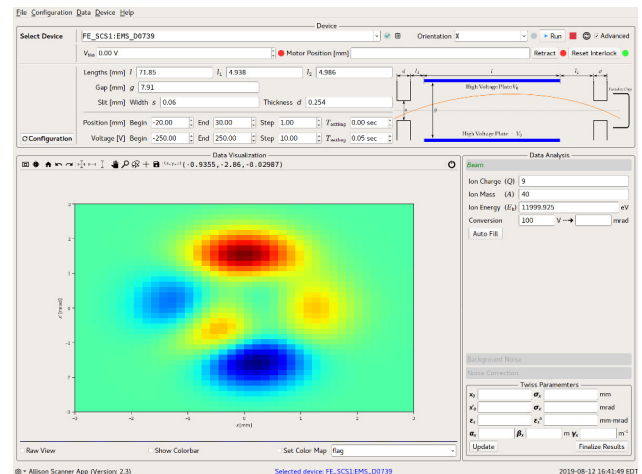


Figure 1: Main window of Allison scanner application developed by PyQt5\*.

\*The image shown here is just a place holder from the widget

While after the data acquisition (DAQ) of the beam transverse phase space, post data analysis is the most important part to get the beam Twiss parameters [7], say  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\epsilon$ , as well as projected beam sizes and center positions. To accomplish this goal, another dedicated GUI application is developed within the software framework addressed in ref [5].

Figure 1 shows the main window of Allison scanner app, from which, the user could have the controls to the device,

\* Work supported by the U.S. Department of Energy Office of Science under Cooperative Agreement DE-SC0000661, the State of Michigan and Michigan State University.

<sup>†</sup> zhangt@frib.msu.edu

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

including set up the motor/voltage scan ranges, as well as other related parameters, e.g. the bias voltage applied onto the integrated Faraday cup, which detects the signal of the deflected ions.

Here are the main features of Allison scanner app:

- Device configuration management;
- Device controls and live DAQ visualization;
- Auto/interactive data post-processing;
- Plenty of controls to the data visualization;
- Save/load data to/from files;

### Device Abstraction

In the framework of PHANTASY [5], all the devices are abstracted into Python objects, from which the devices PVs could be reached by the attributes of Python objects, that way, all the read and write operations could be accomplished by manipulating the Python objects, in any Python terminals, e.g. IPython [8] or Jupyter-notebook [9].

The physicists and other users do not need to have any knowledge about the PVs, which are usually of very long string names. After abstraction, the device itself is a well-enclosed entity, all the attributes that could be controlled could be implied by pressing TAB in the Python terminals (see Fig. 2).

```

In [1]: from phantasy import disable_warnings
In [2]: disable_warnings()
In [3]: from phantasy import MachinePortal
In [4]: mp = MachinePortal('FRIB', 'LEBT')
In [5]: ems = mp.get_elements('type='EMS')[0]
In [6]: ems
Out[6]: FE_SCS1:EMS_D0739 [EMS] @ sb=73.949557
In [7]: ems.
ems.ABORT_SCAN1      ems.CNT_POS1      ems.current_setting
ems.ABORT_SCAN2      ems.CNT_POS2      ems.DATA1
ems.active           ems.CNT_VOLT1     ems.DATA2
ems.BIAS_VOLT        ems.CNT_VOLT2     ems.design_settings
ems.BIAS_VOLT_ON    ems.convert       ems.ENABLE_SCAN1
    
```

Figure 2: Screenshot of the interactive control of Allison scanner device in IPython.

The Allison scanner app is designed to be able to handle similar type devices. Thus the hardware configuration for each device is maintained by static files, switching to a different device name in the *Select Device* dropdown menu should target the corresponding device, whose name should show up in the bottom status bar (see Fig. 1). Figure 3 shows the typical configuration for the Allison scanner devices.

The configurations define the geometry information device by device, and keep the nominal device operation settings for the controls variables, which still could be changed in the main UI.

### Data Visualization

As one of the physics high-level GUI applications developed for FRIB accelerator system, Allison scanner app

```

1 [FE_SCS1:EMS_D0739]
2 info = Installed
3 # unit: [mm]
4 # length: E-dipole plate length
5 # length1: slit-entrance to plate
6 # length2: plate to slit-exit
7 # gap: distance between plates
8 # slit_width: slit width
9 # slit_thickness: slit thickness
10 length = 71.85
11 length1 = 4.938
12 length2 = 4.986
13 gap = 7.91
14 slit_width = 0.06
15 slit_thickness = 0.254
16
17 # bias volt threshold, [V], must be <= this value
18 bias_volt_threshold = -200.0
19
20 [FE_SCS1:EMS_D0739.X]
21 # device scan config, unit: pos [mm], volt [V], time [sec]
22 pos_begin = -15.0
23 pos_end = 25.0
24 pos_step = 1.0
25 pos_settling_time = 0.0
26 volt_begin = -500.0
27 volt_end = 500.0
28 volt_step = 20.0
29 volt_settling_time = 0.05
30
31 [FE_SCS1:EMS_D0739.Y]
32 # device scan config, unit: pos [mm], volt [V], time [sec]
33 pos_begin = -15.0
34 pos_end = 25.0
35 pos_step = 1.0
36 pos_settling_time = 0.0
37 volt_begin = -500.0
38 volt_end = 500.0
39 volt_step = 20.0
40 volt_settling_time = 0.05
    
```

Figure 3: Screenshot of configuration file for device management.

takes advantage of the homemade data visualization widgets for PyQt5 apps. matplotlibImageWidget is for generic image data visualization, one can design the UI by drag&drop the widget-of-interest to the app in Qt-designer [10] (see Fig. 4), all these are made possible by another Python package named mpl4qt, which is built on top of matplotlib [11] to facilitate the modularized Qt widgets for PyQt5 applications [12].

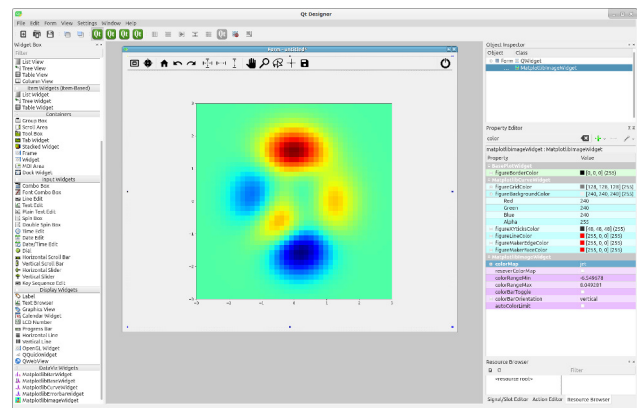


Figure 4: Qt-designer with customized data visualization widgets.

After pushing the *Run* button, Allison scanner device will be starting to move, every step the motor advances, voltage sweeping happens in the defined scan range, then the image plot widget will update with the acquired data. Once the movement is done, the data will be automatically processed, beam Twiss parameters will be on a pop-up window (see Fig. 5). The users also have the chance to interactively adjust the parameters to refine the data processing procedure.

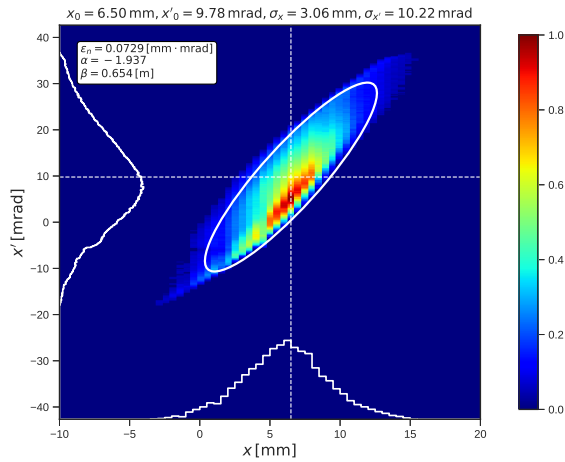


Figure 5: Analyzed Twiss parameters.

The `matplotlibImageWidget` supports online configurations of the image plot, e.g. change the color map, color range, show the color bar, and other adjustment to the figure, via the configuration context menu invoked by right-clicking on the figure canvas. More details could be found in the documentation of `mp14qt` [12].

### App Development and Deployment

The app is developed with Python 3.x [13] and PyQt5 [14], the channel access of EPICS is interfaced by `pyepics` [15]. During the development, a software IOC was developed to simulate the real device behavior, the Allison scanner app can work at simulation mode by issuing the command `allison_scanner --mode simulation`, provided that the soft-IOC is running. Under simulation mode, all acquired data could be analyzed without any issue.

When the development reaches a new milestone, new release is tagged and the source code will be packaged into a Debian package, then automatically deployed by continuous delivery system [16]. Allison scanner app is one of the apps that packaged into a package named `phantasy-apps`, which also contains the development of other FRIB physics high-level applications.

### Working Examples

Here show some working examples for the FRIB LEBT beam, which is  $^{40}_{18}\text{Ar}^{9+}$ , with the kinetic energy of approx 12 keV/u, the distribution of the transverse phase space depends on the beam optics started from the ion source. Figure 6 shows the measured phase space at  $x$  plane, Fig. 7 shows the boundary of the signal and noise (right subfigure), which could be adjusted by drawing on the left subfigure by lasso-tool on the top toolbar.

Once the selected noise/signal boundary is confirmed, one can move back to the *Noise Correction* tab of the main window, to apply the noise correction algorithm, then click

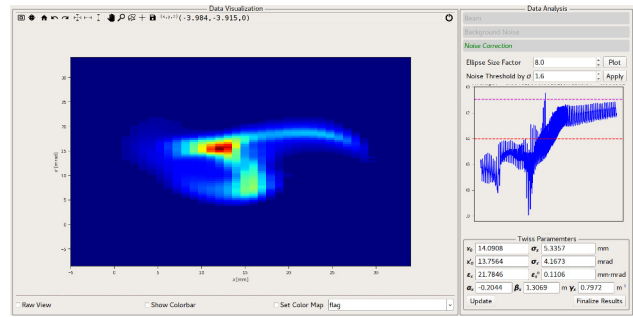


Figure 6: Measure the  $^{40}_{18}\text{Ar}^{9+}$  transverse emittance.

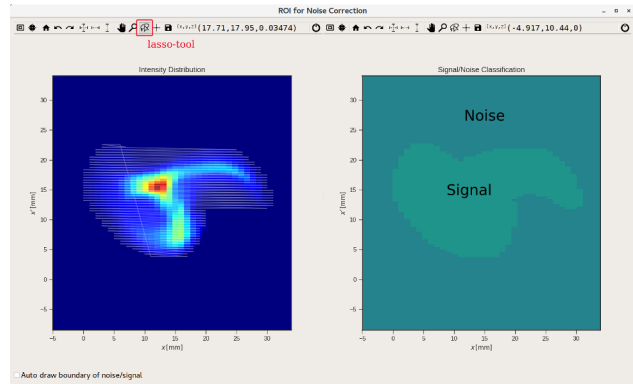


Figure 7: Signal/noise boundary.

the *Update* button to calculate Twiss parameters, the finalized results could be reached via *Finalized Results* button.

It's worth to mention that the arbitrary boundary selection function sometimes could be very useful for the multi-charge states measurement.

All the data, including the acquired raw data and analyzed results could be saved as a file by *File→Save* or CTRL+S.

## CONCLUSION

At FRIB, we developed one dedicated GUI application for Allison scanner to measure the transverse emittance in an efficient way. In this paper, we overview the software development and the experiment results. On the other hand, most of the development could also be able to benefit the community, e.g. a new Allison-Scanner device with the new configuration with expected minimal adjustment could work with this app, and the generic data visualization widgets could be used on any other PyQt5 apps.

## ACKNOWLEDGMENTS

The authors would like to thank K. Fukushima, T. Yoshimoto, A. Plastun, Q. Zhao and P. Ostroumov for the useful discussions on the beam physics, B. Martins, D. Omitto, S. Cogan and S. Lidia for the useful discussions on the beam diagnostics and the help from EPICS community to resolve the issues that happened during the development.

## REFERENCES

- [1] J. Wei *et al.*, “The FRIB Superconducting Linac - Status and Plans”, in *Proc. 28th Linear Accelerator Conf. (LINAC’16)*, East Lansing, MI, USA, Sep. 2016, pp. 1–6. doi:10.18429/JACoW-LINAC2016-M01A01
- [2] J. Wei *et al.*, “Advances of the FRIB project”, *Int. J. Mod. Phys. E*, vol. 28, no. 3, p. 1930003, 2019.
- [3] P. W. Allison, J. D. Sherman, and D. B. Holtkamp, “An emittance scanner for intense low-energy ion beams”, *IEEE Trans. Nucl. Sci.*, vol.30, p. 2204, 1983.
- [4] J. C. Wong, S. M. Lund and T. Maruta, “High resolution phase space measurements with Allison-type emittance scanners”, *Phys.Rev.Accel.Beams*, vol. 22, p. 072801, 2019.
- [5] T. Zhang, “Physics high-level applications and toolkit for accelerator system”, in *EPICS Collaboration Meeting*, Argonne National Laboratory, IL, USA, Jun. 2018. <https://epics.anl.gov/meetings/2018-06/talks.html>
- [6] <https://epics-controls.org/>
- [7] E.D Courant and H.S Snyder, “Theory of the alternating-gradientsynchrotron”, *Annals of Physics*, vol. 3, no. 1, 1958.
- [8] <https://python.org/>
- [9] <https://jupyter.org/>
- [10] <https://doc.qt.io/qt-5/qt designer-manual.html>
- [11] <https://matplotlib.org/>
- [12] <https://phantasy-project.github.io/mpl4qt/>
- [13] <https://www.python.org/>
- [14] <https://www.riverbankcomputing.com/static/Docs/PyQt5/>
- [15] <https://cars9.uchicago.edu/software/python/pyepics3/>
- [16] M. Konrad, “Automatic deployment of EPICS IOCs”, in *EPICS Collaboration Meeting*, Argonne National Laboratory, IL, USA, Jun. 2018. <https://epics.anl.gov/meetings/2018-06/talks.html>